LA-UR-97-4237

Title:	**MCNP4B™ in Parallel on PC's Running Linux**
Author(s):	Tim Goorley
Submitted to:	http://lib-www.lanl.gov/cgi-bin/getfile?00818898.pdf

Los Alamos
NATIONAL LABORATORY
research note

The Applied Theoretical Physics Division
Transport Methods Group

To/MS: Distribution
From/MS: Tim Goorley
Phone/FAX: 665-3682
Symbol: XTM-RN (U) 97-028
Date: July 22, 1997

MCNP4B ã in Parallel on PC's Running Linux

Table of Contents

1.0 Introduction

The new release of MCNP[1], MCNP4B, has the capability of running in parallel on a heterogeneous network of computers running Linux, SUN OS, SUN Solaris and others. Linux is an attractive option since it is free and can be installed on personal computers using Pentium and Pentium Pro chips. When used in parallel, a group of these machines offers a cost effective alternative with much computing power.

Linux offers users much flexibility. It can be obtained free of charge from a variety of distributors, notably Slackware and Red Hat. Linux will run on a variety of hardware: 386, 486, Pentium, Pentium Pro and more. It can be installed on systems already running popular operating systems, such as Windows NT, Windows 95, DOS and others. In this particular demonstration, Slackware Linux 3.2 was installed on a 200 MHz Pentium Pro Micron tower and a 166 MHz Pentium Sager laptop. Red Hat Linux 4.2 was installed on another 200 MHz Pentium Pro Micron tower. Both Microns were running Windows NT, while the laptop was running Windows 95.

2.0 Installation of Linux

Linux installation is reasonably straightforward. First the Linux installation files are obtained from the distributor, commonly Red Hat or Slackware. Included are two files necessary to boot Linux from a pair of floppies. Using the computer's native operating system, the hard drive's partitions are resized, creating free space. The computer is then restarted and Linux is booted from the two floppies. The Linux operating system on these disks is used to create Linux partitions on the hard drive, install the Linux information from CD-ROM or other hard drive partitions, and then reconfigure the operating system to reflect the current hardware. A more complete review of the installation process than what follows can be found in The Linux Installation HOWTO.

Although the Linux operating system contained in the installation files is the same, the different distributions vary drastically in organization. Slackware is organized in installation disk sets, which contain all appropriate files for a full or partial Linux installation. Even though these disk sets represent Linux installation from floppy disks, if the file structure is preserved, the files can be placed on hard drives, CD-ROM or jaz drives. The following disk sets are strongly recommended for Linux, PVM and MCNP installation:

Set A:	General	9 disks
Set AP:	Applications	5 disks
Set D:	Program Development	13 disks
Set E:	Emacs 19.25	8 disks
Set F:	FAQ	3 disks
Set K:	Kernel Source Code	6 disks
Set N:	Networking	6 disks
Set X:	X-Windows	23 disks
Set XA:	X-Windows Applications	4 disks

[1] MCNP is a trademark of the Regents of the University of California, Los Alamos National Labratory.

These disk sets are located on the Slackware CD-ROM, or they can be binary FTP'ed free of charge from ftp.cdrom.com:/pub/Linux/Slackware. If FTP is used, each floppy disk of information must be transferred into its own appropriate subdirectory. The entire process of downloading the above disk sets using the computer's native operating system onto a FAT partition took three to four hours. The download was performed using Windows NT. Had the partition been formatted NTFS, which Linux has difficulty recognizing, the disk sets would need to be copied to a FAT jaz drive or the CD-ROM should be ordered. Red Hat does not use this organization scheme.

The Linux operating system essential files are small enough to fit onto two floppy disks, called the boot and root disks. A boot disk is created from a boot disk image which is specific to categories of computers: those with IDE or SCSI hard drives, those with certain kinds of CDROM drives, etc. The Flashpoint SCSI card installed on two Micron machines required a special boot image, buslogic.bin, found at http://www.dandelion.com/Linux/. After downloading, buslogic.bin was then be copied onto a floppy disk from a DOS system with the command "rawrite". In UNIX, the command "dd" is used: "dd if=buslogic.bin of=/dev/fd0 obs=18k". To create a boot disk for the Windows 95 laptop, a boot disk that can recognize the FAT32 IDE hard disk, FAT32.i, was copied. Two generic boot disks are scsi.s and bare.i. In a similar fashion, color144.gz and Pcmcia.gz were also copied onto floppies for the Microns and laptop root disks respectively. Many other boot disk and root disk images can be found at the Linux installation FTP site, ftp.cdrom.com:/pub/linux, as well as the files that describe which images are appropriate for specific hardware. These boot and root floppies also serve as emergency repair disks.

Before a full installation can proceed, the hard drive should be properly prepared. There should be at least one gigabyte of free space on the computer's hard drive to install Linux, PVM and MCNP. Using a program called Partition Magic, it is possible to resize the existing FAT, FAT32 or NTFS hard drive partitions. Unfortunately, Linux can only have read-write capability with FAT and FAT32 formats, and the hard drive format cannot be changed without loosing the data. After this free space is created, the computer is booted using the Slackware or Red Hat installation disks. After logging in as root, fdisk is started either manually or automatically, depending on the Linux distribution. Fdisk can create, destroy and modify the hard drive's partitions. By typing "n" at the prompt, a new partition is created from the free space. This partition should start in the next consecutive sector from the previous partition, but should not take up the remaining space. In the last 100 megabytes, a second new partition should be created for the Linux swap space. The file systems should be specified as Linux native and Linux swap, 83 and 82, by typing "t". Finally, "w" should be specified to write the changes to the disk and exit.

Linux installation follows automatically for Red Hat users, while Slackware installers must type "setup". Both guide the user. After Linux is installed, the kernel can be remade for its specific machine configuration by typing "/usr/src/linux make config". After answering many questions concerning hardware and other options, the kernel will be rebuilt. New boot disks can be

created by typing "make zDisk" in the /usr/src/linux directory. If LILO is installed on the hard drive, "make zlilo" will update the boot program so that the new kernel will load on the next reboot.

REDHAT vs. SLACKWARE

Linux comes from a variety of distributors, notably Red Hat and Slackware. Although the two companies distribute a Linux operating system very similar, the user installation procedures are completely different. Red Hat is known for its easier graphical interface, but Red Hat Linux 4.2 was more difficult to install than Slackware Linux 3.2. Red Hat had a bug in the password writing routine, preventing normal login after completing the setup. Although the solution was to type "Linux single" from the LILO boot prompt and erase the second field of the root user in /etc/passwd, this is certainly not intuitive. Another drawback is that the Red Hat network configuration program runs in X Windows. On several laptop computers, where the display hardware does not support X windows, it would have been much more difficult to configure the network. Although Slackware does require a more adept installer, it offers more flexibility in what applications are installed and in what way.

GNU COMPILERS

Although both distributions of Linux came with a C compiler, the newest versions of GNU C and GNU F77 were installed. They can be obtain via anonymous FTP to prep.ai.mit.edu in the /pub/gnu directory. GCC 2.7.2.2 and G77 0.5.20 were installed in the /usr/local/src directory. The following is the installation process:

1. mv g77-0.5.20.tar.gz /usr/local/src; mv gcc-2.7.2.2.tar.gz /usr/local/src
2. cd /usr/local/src
3. gunzip g77-0.5.20.tar.gz ; gunzip gcc-2.7.2.2.tar.gz
4. tar -xf g77-0.5.20.tar.gz ; tar -xf gcc-2.7.2.2.tar.gz
5. ln -s gcc-2.7.2.2 gcc ; ln -s g77-0.5.20 g77
6. mv -i g77/* gcc
7. patch -p1 -d gcc-2.7.2.2 < gcc-2.7.2.2./f/gbe/2.7.2.2.diff
8. cd gcc
9. touch f77-install-ok ; touch f2c-install-ok
10. ./configure -- prefix=/usr –host=i585-*-linuxlibc1
11. make bootstrap
12. rm -fr stage1
13. make -k install
14. g77 -v ; gcc –version

After the compilers are created, the Linux X Windows server, X11R6 can be configured. This process is dramatically different for the different Linux distributions Although not necessary, X11R6 allows the user to run many graphical programs, including the plotting capabilities of MCNP. After installation, the server can be started by typing "startx".

3.0 Installation of PVM

Parallel Virtual Machine installation is comprised of three simple steps. After obtaining the most recent version of PVM, 3.3.11, available by anonymous FTP to netlib2.cs.utk.edu, it should be unzipped and untarred in the /usr/local/src directory. The PVM environmental variables then need to be set by typing "export variable=value" or "setenv variable value", depending on whether the command shell is bash or csh. PVM_ROOT should be set to "/usr/local/src" and PVM_ARCH should equal "LINUX". In the pvm3 directory created by untarring the file, make is typed to create the libraries and executables. Several PVM programs provided in the distribution can be compiled and used to test the virtual machine. For example, hello is created by typing "aimk hello" in the examples directory.

Once a working PVM executable is created, proper network connectivity should be ensured. The /etc/hosts.allow needs to allow rsh connections from other computers. Remote hosts also need to be added by name to the .rhosts file. Two programs, rsh and rlogin need to be working properly before PVM can add remote hosts to the virtual machine. The permissions of these programs should be set to -rwsr-xr-x and should be owned by root. The root user, however, will be unable to use them. Other users may need to add the full name, ip address, local name and aliases of all remote hosts and the local host to the .rhosts file located in their home directory. If the local host still cannot add remote machines, it may be useful to check if environmental variables are being passed correctly. Type:

```
rsh remote_host '$PVM_ROOT'/lib/pvmd -s
```

The response should be:

```
[pvmd pid6918] slave_config: bad args
[pvmd pid6918] pvmbailout(0)
```

Another environmental variable, PVM_DPATH may need to be set to pvm3/lib/pvmd on the local Linux machine to add SUN or Solaris machines, provided the files are in $HOME/pvm3 on the remote system. Even if hosts cannot be added from the PVM console, it is possible to manually startup and connect the PVM deamons. The PVM manual is a useful reference for trouble shooting.

For MCNP to run in parallel, several criteria must be satisfied. Besides the requirements for PVM to run, MCNP must be built with the PVM option and linked with the PVM libraries. A soft link to the MCNP executable, called mcnp.pvm, must exist in the $HOME/pvm3/bin/PVM_ARCH directory on every remote machine.

4.0 Installation of MCNP4B

MCNP4B can be installed on Linux systems in a similar manner to other systems, as described in appendix C of the MCNP manual. The install.fix file that contains both Linux and PVM patches is needed for the PVM enabled version of MCNP, as well as a new version of prpr.id.

The Linux install.fix file contains the Linux compile options under the sixth operating system, DEC UNIX. Below is the relevant portion of the file.

```
********** FIXES FOR SYSTEM 6 (DEC UNIX and PC Linux) **********
c
c     The following 7 lines enable GNU F77 compilation on Linux.
6 1 2 1 1
*define linux
6 2 27 3 1
  -O0
6 2 28 4 28
6 3 2 1 1
*define linux
```

The first and last line add the Linux option to take advantage of a Linux patch at the end of the install.fix file. The second line adds some compiler options to f77 -c *.f, while the third deletes the line f77 -O1 -fpe1 -c itally.f getxs.f gmgww.f. The Linux patch at the bottom of the install.fix file is:

```
*/ ------------------------------------------------------------ getidt
*/ Provide cleaner separation of Linux and DEC.      06/04/97  (GWM/GWM)
*d,gi4b.1                                          <42208>
*if def,linux,3
*d,gi4b.5                                          <42212>
*i,gi4a.1                                         <42213>
*if -def,linux,3
```

The install.fix files are available by FTP to CFS.

MCSETUP

The installation process compiles and runs MCSETUP, which creates a batch file that will compile PRPR, MAKXSF and MCNP. Several options in MCSETUP must be changed to create the appropriate MCNP executable. The computer system description, option 1.1, must be changed from Cray Unicos to dec. The dynamic memory option, since pointer creates many compile errors in Linux, must be avoided by setting the mdas value under option 2.3. If mcplot and plot are used, the X Windows Library, option 3.3, must be changed to find the corresponding Linux X11R6 files The user is directed to enter the library directory, /usr/X11R6/lib, the library, libX11.so, and the include directory, /usr/X11R6/include. The cross section data was installed in /usr/mcnp4b/xs, which can be entered in MCSETUP as option 4.1. The PVM libraries necessary to run the distributed memory version of MCNP, option 5.1, are located in /usr/local/src/pvm3/lib/LINUX. The default names, libfpvm3.a and libpvm3.a are correct.

PRPR

MCSETUP will create a batch program called makemcnp, which compiles PRPR. PRPR is the preprocessor used to combine patch files and the MCNP4b source code, mcnp4b.id. While trying to compile prpr.id, f77 gave the following memory warning: "Initialization of large (216000-unit) aggregate area 'hc' at (^) currently very slow and takes lots of memory during g77 compile – to be improved in 0.6". While PRPR was compiled and linked after several minutes, a simple change allows compilation in two or three seconds.

```
****** prpr.orig
     3 hj/'define','comdeck','deck','call','if','endif'/
       data hc/lc*' '/,hi/' '/,hl/' '/,hn/kc*' '/,hq/2*' '/,
     1 hs/40*' '/,ht/' '/,hz/' '/
****** prpr.new
     3 hj/'define','comdeck','deck','call','if','endif'/
       data          hi/' '/,hl/' '/,hn/kc*' '/,hq/2*' '/,
     1 hs/40*' '/,ht/' '/,hz/' '/
******

****** prpr.orig
       data i4,i5,ic,il,ik,ja,jd,kd,lq,ns/0,1,9*0/
       nc(1)=1
****** prpr.new
       data i4,i5,ic,il,ik,ja,jd,kd,lq,ns/0,1,9*0/
       do 5 i=1,lc
     5 hc(i)=' '
       nc(1)=1
******
```

5.0 Conversion of Cross Sections

Many users will want to convert their type one ASCII cross sections to type two binary files, due to the considerable disk space reduction. The MAKXSF program, which does this conversion, is compiled in the makemcnp script. This program is described in appendix C of the MCNP manual. A file called specs needs to be changed before the cross sections can be converted. The numbers 2048 and 512, representing the record length and entries per record respectively, need to be added to every line except the last, which uses a recl number of 4096. A Linux specs file would look like:

```
xsdir1   xsdir2
endf601  endf602  2 2048 512

newxs1     newxs2   2 2048 512
……..
mcplib02  mcplb022  2 2048 512

mcplib1    mcplib2  2 2048 512
```

el1 el2 2 4096 512

Using the jaz drive is a convenient way of converting the cross sections, since more than enough space is available. If the jaz drive is MSDOS formatted, the MSDOS naming rules apply. It is necessary to change mcplib02 to mcplb022, not mcplib022. Had the conversion been carried out under the Linux file system, ex2fs, this change should not be necessary. MAKXSF was able to convert the 569 megabytes of ASCII library files to 94 megabytes of binary files.

6.0 Verification of MCNP4B

The test suite of twenty-nine test problems is used for MCNP verification. The scripts runprob and runprobmt verify the standard and PVM versions. Both of these scripts should reference the DOS MCTL files, which need to be uncompressed on a DOS machine and FTP'ed to the Linux system. These scripts will compare the tally output of the test problems to the reference tally output, the MCTL files, using the command "diff". When MCNP4b is run sequentially, only one non-zero difference file, difm12, is created. It indicates that a different number of random numbers was used during this execution than was used in the reference execution. This inconsistency is attributed to differences in the Layhey F77 and GNU F77 compilers. When run using PVM, four difm files will be created. Two of them, inp12m and inp08am, do not track. The other two, inp21am and inp28m, are output differences caused by PVM's rendezvous procedure, which is acceptable . Below is a table summarizing the difference files, difm12, difm21a, and difm28, and the files themselves. Difm08a, beyond the header, is too extensive to include here.

Table 1: Non-Zero Difm Files from MCNP4B on Linux

-O0 Optimization	w/o PVM	(runprob)	w/ PVM	(runprobmt)
File	Size (bytes)	Error	Size (bytes)	Error
difm08a	N/A	N/A	2669	tracking
difm12	128	tracking	3508	tracking
difm21a	N/A	N/A	1276	rendezvous
difm28	0	none	698	rendezvous

The above files were created when MCNP was compiled with -O0 optimization.

Difm08a header - 2669 bytes

```
1c1
<                    2    1579      21050
---
>                    2    1570      19337
```

Difm12 - 146 bytes

```
1c1
<                    2    3000      4774870
---
>                    2    3000      4774894
```

Difm21a - 1276 bytes

```
273c273,274
```

```
< tfc  10     1     1     22     1     1     1     23     1
---
> tfc  20     1     1     22     1     1     1     23     1
>       1000  4.09176E-05  3.71303E-01
274a276
>       3000  4.08754E-05  1.78175E-01
275a278
>       5000  4.55530E-05  1.68729E-01
276a280
>       7000  4.26109E-05  1.35474E-01
277a282
>       9000  4.09342E-05  1.15212E-01
278a284
>       11000  4.71283E-05  1.23183E-01
279a286
>       13000  4.66970E-05  1.11987E-01
280a288
>       15000  4.50165E-05  1.04542E-01
281a290
>       17000  5.70298E-05  2.23201E-01
282a292
>       19000  5.48663E-05  2.07877E-01
298c308,309
< tfc  10     1     1     1     1     1     1     4     1
---
> tfc  20     1     1     1     1     1     1     4     1
>       1000  4.09176E-05  3.71303E-01
299a311
>       3000  4.08754E-05  1.78175E-01
300a313
>       5000  4.55530E-05  1.68729E-01
301a315
>       7000  4.26109E-05  1.35474E-01
302a317
>       9000  4.09342E-05  1.15212E-01
303a319
>       11000  4.71283E-05  1.23183E-01
304a321
>       13000  4.66970E-05  1.11987E-01
305a323
>       15000  4.50165E-05  1.04542E-01
306a325
>       17000  5.70298E-05  2.23201E-01
307a327
>       19000  5.48663E-05  2.07877E-01
```

Difm28 - 689 bytes

```
475,476c475,477
< tfc  9     1     1     42     1     1     1     43     1
<       3000  4.92945E-05  6.68206E-01
---
> tfc  13     1     1     42     1     1     1     43     1
>       2000  6.53503E-05  7.50077E-01
>       4000  5.91882E-05  4.71549E-01
478c479,480
```

```
<      9000  6.80561E-05  3.98976E-01
---
>      8000  7.29853E-05  4.17453E-01
>     10000  6.23468E-05  3.92323E-01
480c482,483
<      15000  5.00067E-05  3.34972E-01
---
>      14000  5.30353E-05  3.38257E-01
>      16000  6.61325E-05  3.71568E-01
482c485,486
<      21000  6.19494E-05  3.09465E-01
---
>      20000  5.87871E-05  3.36357E-01
>      22000  5.99188E-05  3.05517E-01
```

The above differences were created with a compilation of MCNP with out optimization. Although zero optimization produces the fewest number of difm files, runtimes can be considerably reduced at higher optimizations. When the -O2 optimization is enabled, the following difm files are created when runprob and runprobmt are run:

Table 2: Non-Zero Difm Files from MCNP4B on Linux

-O2 Optimization	w/o PVM (runprob)		w/ PVM (runprobmt)	
File	Size (bytes)	Error	Size (bytes)	Error
difm08a	N/A	N/A	2669	tracking
difm12	3508	tracking	3508	tracking
difm21a	N/A	N/A	1276	rendezvous
difm28	0	none	698	rendezvous
difm29	1686	tracking	1686	tracking
difm29a	N/A	N/A	1686	tracking

The above files were created when MCNP was compiled with -O2 optimization.

There are also CPU specific optimizations. The -m486 option is optimization for the 486. The optimization for the Pentium is: -O2 -m486 -malign-loops=2 -malign-jumps=2 -malign-functions=2. There currently is no specific optimization for the Pentium Pro. The difference between the 486 options and the Pentium options on a Pentium Pro machine are negligible. Since optimizations above -O0 cause more problems not to track, they clearly change answers and thus are officially unacceptable.

Linux is a powerful operating system that can be installed on a variety of personal computers. MCNP4B can be installed on Linux machines and run in parallel over a heterogeneous network of computers. Installation is straight foreward and can be completed by running "./install dec" with Linux install.fix and prpr.id files, along with the other required files. The quality of the program can be verified by running the test suite, which produces two tracking errors, potentially attributed to compiler differences. The result is an officially acceptable version of MCNP4B.

Acknowledgment

I would like to thank Dr. Gregg McKinney for his time, effort and extensive knowledge. Without his help and a previous memo by Cameron Kellough, this project would have been much more difficult.

REFERENCES

1. J.F. Briesmeister, Ed., "MCNP - A General Monte Carlo N-Particle Transport Code," v. 4B, Los Alamos National Laboratory report LA-12625-M Version 4B (March 1997).

2. Al Geist, et. all. "PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing" Cambridge, MA. The MIT Press. 1994

Appendix A - Linux PVM Install.fix

```
c        <<<<< makemcnp changes for MCNP4B >>>>>       loddat 07/07/97
c
c   For each fix, enter five integer parameters on one line in free
c   format, ip(1)-ip(5) described below, followed by the appropriate
c   number of new lines.
c
c   ip(1) = The applicable computer system number or 0 for all systems.
c   ip(2) = The file number:
c       1 = patchf
c       2 = makemcnp
c       3 = patchc
c   ip(3) = The applicable line number to edit-see the MCSETUP source.
c   ip(4) = The alter code:
c       1 = insert the following ip(5) line(s) before line ip(3)
c       2 = replace line ip(3) with the following ip(5) line(s)
c       3 = insert the following line after entry ip(5) of line ip(3)
c       4 = delete lines ip(3) through ip(5)
c   ip(5) = The number of new line(s) if ip(4)=1,2
c           The entry number if ip(4)=3
c           The last line number to delete if ip(4)=4
c
c   The number of new lines that follow ip(1)-ip(5) is given by:
c         = ip(5) if ip(4)=1,2
c         = 1 if ip(4)=3
c         = 0 if ip(4)=4

c******************** FIXES FOR SYSTEM 1 (Cray UNICOS) ******************
c
c    Add the *define t3d for the Cray T3D.
c     1 1 2 1 1
c    *define t3d
c
```

```
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     Change the CFT77 compile line.
c     1 2 18 1 4
c     export TARGET; TARGET=CRAY-T3D
c     ln -s /mpp/bin/cc cc
c     ln -s /mpp/bin/cft77 cft77
c     ln -s /mpp/bin/mppldr segldr
c     Provide links to the T3D compilers and linker. XTM:GWM-95-128
c     1 2 26 2 1
c     cft77 compile
c
c     Add CFTLIB library on UNICOS for profile timing.    06/30/95 (GWM)
c     1 2 28 3 10
c      -L/usr/local/lib -lprof

c********************* FIXES FOR SYSTEM 2 (Sun  SunOS ) *****************

c     Add a second X-window include path for some SunOS systems.
c     2 2 18 3 4
c      -I/usr/openwin/include/X11

c     Add a link to the ANSI C library (libansi.a) on some SunOS systems.
c     2 2 29 3 10
c      -L/home -lansi

c********************* FIXES FOR SYSTEM 3 (IBM  AIX   ) *****************
c
c     PVM libraries must go first for pvm version at some installations.
c     3 2 30 2 1
c  xlf -o mcnp *.o -L/usr/lanl/pvm3.3/lib -lfpvm3 -lpvm3 -L/usr/lib -lX11

c********************* FIXES FOR SYSTEM 4 (HP   HPUX  ) *****************
c
c     Add a link to the C math library (libM.a) on some HP systems.
c     4 2 29 3  6
c      -lM

c********************* FIXES FOR SYSTEM 5 (VAX  VMS   ) *****************
c
c     Add an additional *define keyword for PRPR on a VAX VMS system.
c      5 1 2 1  1
c     *define disscgs

c***************** FIXES FOR SYSTEM 6 (DEC UNIX and PC LINUX) ************
```

```
c
c     The following 7 lines enable GNU F77 compilation on LINUX.
c     Tim Goorley, jgoorley@mit.edu XTM-RN(U97-028) 7/21/97
c     6 1 2 1 1
c     *define linux
c     6 2 27 3 1
c      -O0
c     6 2 28 4 28
c     6 3 2 1 1
c     *define linux

c********************** FIXES FOR SYSTEM 7 (PC   DOS   ) ******************
c
c     The following 9 lines enable LAHEY F90 compilation on DOS/WINDOWS.
c     Also uncomment the 2 lines in TTYINT below [call break()].
c     7 2 8 2 1
c     lf90 prpr.for
c     7 2 9 4 9
c     7 2 17 2 1
c     lf90 makxsf.for
c     7 2 18 4 18
c     7 2 43 2 1
c     lf90 mcnp.for
c     7 2 50 4 50

c********************** FIXES FOR SYSTEM 8 (Sun  Sol.  ) ******************

c********************** FIXES FOR SYSTEM 9 (SGI  IRIX  ) ******************


c*********** FIXES FOR THE MCNP FORTRAN SOURCE (PATCHF FILE)
************

0 1 11 2
*/ Integrate your patch with the following. If your patch makes changes
*/ to the ZC, VV, CM, GS, MB, or BD decks, contact MCNP@LANL.GOV for
*/ details on including your patch via the INSTALL.FIX file. When
*/ integrating your patch, be sure the order of the changes (increasing
*/ in line number) is preserved to avoid a PRPR error.  See Appendix
*/ C, page C-8 in the MCNP manual for more details.
*/
*/ ------------------------------------------------------------ ttyint
*/ Uncomment (delete 1st 3 columns) of next 2 lines to enable LAHEY F90
*/ compilation on DOS/WINDOWS.                  07/07/97  (GWM/GWM)
```

```
*/ *d,tt4a.28                                                    <1014>
*/      call break(lockl)
*/
*/ ------------------------------------------------------------ msgcon
*/ Fix a print bug.  Wrong number of arguments.      07/07/97  (GWM/GWM)
*d,me4b.195                                                      <20311>
      call errprn(0,j,1,one*max(1,ntasks),zero,'ntk',' ',
*/ Force PVM to free some buffers.                  07/07/97  (GWM/GWM)
*/ $20 awarded to Dudley A. Raine, III, ORNL (XTM:JSH-97-176)  (GWM/GWM)
*i,me4b.311                                                      <20498>
c
c       free the receive buffers.
      do 582 nt=1,ntasks
  582 call mfbuf(ib(nt),i)
*/
*/ ------------------------------------------------------------ startp
*/ Wrong index. Affects var. reduction and tracking. 07/07/97  (GWM/GWM)
*d,sp4b.8,sp4b.10                                                <21159-21161>
      if(wc1(i).ge.0.)go to 85
      wcs1tc(i)=-wc1(i)*wgt
      wcs2tc(i)=-wc2(i)*wgt
*/
*/ ------------------------------------------------------------ ypbssp
*/ Wrong index. Affects var. reduction and tracking. 07/07/97  (GWM/GWM)
*d,yp4b.2,yp4b.4                                                 <22590-22592>
      if(wc1(i).ge.0.)go to 45
      wcs1tc(i)=-wc1(i)*wgt
      wcs2tc(i)=-wc2(i)*wgt
*/
*/ ------------------------------------------------------------ mfbuf
*/ Force PVM to free some buffers.                  07/07/97  (GWM/GWM)
*/ $20 awarded to Dudley A. Raine, III, ORNL (XTM:JSH-97-176)  (GWM/GWM)
*i,mf4b.83                                                       <41720>
      subroutine mfbuf(ib,in)
c       perform function 'free message buffer'.
*call mb
*if def,pvm,1
      call pvmffreebuf(ib,in)
      return
      end
*/
*/ ------------------------------------------------------------ getidt
*/ Provide cleaner separation of LINUX and DEC.      06/04/97  (GWM/GWM)
*d,gi4b.1                                                        <42208>
```

```
*if def,linux,3
*d,gi4b.5                                    <42212>
*i,gi4a.1                                    <42213>
*if -def,linux,3
```

Distribution:

R.J. Juzaitis, X-DO, B218
W.H. Reed X-DO, B226
P.D. Soran, XTM, B226
D.G. Shirk, XTM, B226
K.J. Adams, XTM, B226
T.E. Booth, XTM, B226
J.F. Briesmeister, XTM, B226
G.P. Estes, XTM, B226
R.A. Forester, XTM, B226
J.S. Hendricks, XTM, B226
G.W. McKinney, XTM, B226
XTM File